UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/772,698	02/05/2004	Gansha Wu	ITL.1069US (P18137)	4898

21906 7590 07/27/2006

TROP PRUNER & HU, PC
1616 S. VOSS ROAD, SUITE 750
HOUSTON, TX 77057-2631

EXAMINER	
SYED, FARHAN M	
ART UNIT	PAPER NUMBER
2165	

DATE MAILED: 07/27/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

	Application No.	Applicant(s)	
Office Action Summary	10/772,698	WU ET AL.	
	Examiner	**Art Unit**	
	Farhan M. Syed	2165	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _05 February 2004_.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-28_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-28_ is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _05 February 2004_ is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _20040601_.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-28 are pending.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claim 13 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

4. Claim 13 recites the limitation "the requestor" in line 2. There is insufficient antecedent basis for this limitation in the claim.

5. Claims 18-24 and 26-28 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential elements, such omission amounting to a gap between the elements. See MPEP § 2172.01. The omitted elements are: It is unclear to the Examiner how the instructions, as recited on line 1 of each claim, are executed to enable to enable the system to perform some limitations as further recited in these claims. The Applicant needs to clearly state how these instructions are to be executed. .

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1 and 18 are rejected under 35 U.S.C. 101 because the claimed invention lacks patentable utility. As per claims 1 and 18, the limitations of each claim recite storing method metatadata including a cookie indicator in a code portion may constitute a mental step that does not create or provide a tangible result that may provide some utility.

Claims 18-24 and 26-28 are rejected under 35 U.S.C. 101 because the disclosed invention is inoperative and therefore lacks utility. The limitations in these claims recite, "if executed" on line 1 of each claims, whereby when the if statement is not executed, the limitations of these claims would either be inoperable or would not create a tangible result.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 1, 2, 5, 7, 10, 12, 13, 14, 16-18, 21, 23, 24, 27, and 28 are rejected under 35 U.S.C. 102(e) as being anticipated by Boger et al (U.S. Patent 6,996,556 and known hereinafter as Boger).

As per claims 1 and 18, Boger teaches a method comprising: storing method metadata including a cookie indicator in a code portion (i.e. *"The embodiments discussed hereinafter utilize a metadata manager to facilitate the collection of metadata from a database, and to supply such metadata to a plurality of optimizer instances in connection with the optimization of database queries." "As an additional matter, by implementing metadata handling functionality outside of optimizer instances, improvements to metadata collection, refinement and maintenance algorithms may be implemented within a metadata manager without requiring modification of optimizer program code."* The preceding text clearly indicates that method metadata are a collection of metadata, a cookie indicator is the optimizer, and code portion is an instance of a program code.)(Column 4, lines 50-54; lines 64-67; column 5, lines 1-2).

As per claim 2, Boger teaches a method, wherein the method metadata further comprises a method handle (i.e. *"As discussed hereinafter, a metadata manager may be configured to handle the tasks of collecting, refining, retrieving and/or maintaining metadata for one or more databases, thus off-loading responsibility for such tasks from individual optimizers or optimizer instances that make use of that metadata."*)(Column 4, lines 54-59).

As per claim 5, Boger teaches a method, further comprising storing the method metadata at an end of the code portion (i.e. *"The embodiments discussed hereinafter utilize a metadata manager to facilitate the collection of metadata from a database, and to supply such metadata to a plurality of optimizer instances in connection with the optimization of database queries." "As an* *additional matter, by implementing metadata handling functionality outside of optimizer instances, improvements to metadata collection, refinement and maintenance algorithms may be implemented within a metadata manager without requiring modification of optimizer program code."*)(Column 4, lines 50-54; lines 64-67; column 5, lines 1-2).

As per claims 7 and 21, Boger teaches a method, further comprising querying the code portion for the method metadata (i.e. *"The primary task of a query optimizer is to choose the most efficient way to execute each database query, or request, passed to the database management system by a user." "An optimizer is often permitted to rewrite a query (or portion of it) into any equivalent form, and since for any given query there are typically many equivalent forms, an optimizer has a countably infinite universe of extremely diverse possible solutions (plans) to consider."*)(Column 1, lines 63-66; Column 2, lines 17-22).

As per claims 10, 23, and 27, Boger teaches a method, further comprising storing the method metadata between a first basic block and a second basic block of the code portion (i.e. *"The embodiments discussed hereinafter utilize a metadata manager to facilitate the collection of metadata from a database, and to supply such metadata to a plurality of optimizer instances in connection with the optimization of database queries."*)(Column 4, lines 50-54; lines 64-67).

As per claim 12, Boger teaches a method further comprising storing the method metadata in a basic block used for exception handling (i.e. *"The embodiments discussed hereinafter utilize a metadata manager to facilitate the collection of metadata from a database, and to supply such metadata to a plurality of optimizer instances in connection with the optimization of database queries."*)(Column 4, lines 50-54; lines 64-67).

As per claim 13, Boger teaches a method comprising: receiving a request to query a code portion for a method bundle including method metadata (i.e. *"FIG. 10 illustrates an exemplary database environment 200 consistent with the invention, subsequent to optimization of three queries 202, 204 and 206 by a plurality of associated optimizer instances 208, 210 and 212."* The preceding text clearly indicates that a request is made to execute three queries.)(Column 15, lines 45-48); searching the code portion for the method bundle (i.e. *"In response to a request for data access descriptors by the optimizer instance, context 224 would determine whether data source caches existed for either of tables T1 and T2, which are accessed by query 202."* The preceding text clearly indicates that once a quest is received, searching the code portion is the process of determining whether data source caches exist for tables that are accessed by the query.)(Column 15, lines 60-63); and returning the method bundle to the requestor (i.e. *"Once the available DAD's are returned by context 224 to optimizer instance 208, the optimizer instance will generate expressions, e.g., expressions 236, 238, 246 and/or 248 as appropriate to access query 202."* The preceding text clearly indicates that results are returned to the requestor, where the results are DADs and the requestor is the optimizer instance.)(Column 16, lines 17-20).

As per claim 16, Boger teaches a method, wherein searching the code portion comprises searching in an instruction cache (i.e. *"In response to a request for data access descriptors by the optimizer instance, context 224 would determine whether data source caches existed for either of tables T1 and T2, which are accessed by query 202."*)(Column 15, lines 60-63).

As per claim 24, Boger teaches a system comprising: a memory including instructions (i.e. *"Computer 10 typically includes a central processing unit (CPU) 12 including one or more* *microprocessors coupled to a memory 14, which may represent the random access memory (RAM) devices comprising the main storage of computer 10, as well as any supplemental levels of memory, e.g., cache memories, non-volatile or backup memories (e.g., programmable or flash memories), read-only memories, etc. In addition, memory 14 may be considered to include memory storage physically located elsewhere in computer 10, e.g., any cache memory in a processor in CPU 12, as well as any storage capacity used as a virtual memory, e.g., as stored on a mass storage device 16 or on another computer coupled to computer 10."* The preceding text clearly indicates that the prior art contains a memory including instructions coupled with microprocessors.)(Column 5, lines 47-59) that if executed enable the system to search a code portion for method metadata including a cookie indicator (i.e. *"In response to a request for data access descriptors by the optimizer instance, context 224 would determine whether data source caches existed for either of tables T1 and T2, which are accessed by query 202."* The preceding text clearly indicates that a search is executed to determine whether data source caches exist.)(Column 15, lines 60-63); a processor coupled to the memory (i.e. *"Computer 10 typically includes a central processing unit (CPU) 12 including one or more microprocessors coupled to a memory 14, which may represent the random access memory (RAM) devices..."* The preceding text clearly indicates that a processor is a microprocessor.)(Column 5, lines 47-50) to execute the instructions (i.e. *"Program code typically comprises one or more instructions that are resident at various times in various memory and storage devices in a computer, and that, when read and executed by one or more processors in a computer, cause that computer to perform the steps necessary to execute steps or elements embodying the various aspects of the invention."* The preceding text clearly indicates that instructions are executed by one or more processors.)(Column 6, lines 40-46); and a wireless interface coupled to the processor (i.e. *"For interface with a user or operator, computer 10 typically includes a user interface 18 incorporating one or more user input devices (e.g., a keyboard, a mouse, a trackball, a joystick, a touchpad, and/or a microphone, among others) and a display (e.g., a CRT monitor, an LCD display panel, and/or a speaker, among others). Otherwise, user input may be received via another computer or terminal, e.g., via a client or single-user computer 20 coupled to* *computer 10 over a network 22."* The preceding text clearly indicates that a wireless interface is a user input received via another computer or terminal.)(Column 5, lines 60-67; column 6, lines 1-3).

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 3, 4, 6, 8, 9, 11, 15, 19, 20, 22, 25, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boger et al (U.S. Patent 6,996,556 and known hereinafter as Boger) in view of Adl-Tabatabai et al (U.S. Patent 6,317,869 and known hereinafter as Adl-Tabatabai).

As per claims 3, 19, and 25, Boger does not explicitly teach a method wherein the method metadata comprises a magic cookie having a bit pattern non-compliant with an instruction set architecture.

Adl-Tabatabai teaches a method wherein the method metadata comprises a magic cookie having a bit pattern non-compliant with an instruction set architecture (i.e. *"However, Java bytecodes have "ambiguous types" where the same variable may hold reference and non-reference values at different times during the execution of the method. A method is essentially a function or procedure in the program."* The preceding text clearly indicates that the non-compliant bit pattern is the non-reference value in the Java bytecodes.)(Column 6, lines 1-3).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Adl-Tabatabai to include a method wherein the method metadata comprises a magic cookie having a bit pattern non-compliant with an instruction set architecture with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claims 4, 15, 20, 26, Boger does not explicitly teach a method, wherein storing the method metadata comprises storing the method metadata at an N-aligned address of the code portion.

Adl-Tabatabai teaches a method, wherein storing the method metadata comprises storing the method metadata at an N-aligned address of the code portion (i.e. *"The memory space 400 in FIG. 4a comprises of a run-time stack 410, registers 450, static variables (462, 468, 472) and objects (460, 464, 466, 470) on the heap. The run-time stack 410 may further comprise of a plurality of activation frames (420, 430, 440) for various program functions and methods. These activation frames are used as working space for functions and methods called during execution of the Java program. During program execution, numerous objects or variables may be declared and used within the program. An object, also referred to as a cell or node, is a run-time notion; any object is an instance of a certain class, created at execution time and made of a number of fields. An object may be a dynamically created class instance or an array or an individually allocated piece of data. An object is assumed to be a contiguous array of bytes or words, divided into fields. A field may contain a pointer or a non-pointer value."*)(Column 4, lines 58-67; column 5, lines 1-5).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Adl-Tabatabai to include a method, wherein storing the method metadata comprises storing the method metadata at an N-aligned address of the code portion with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claims 6 and 11, Boger does not explicitly teach a method, wherein the code portion comprises compiled code for a method corresponding to the method metadata.

Adl-Tabatabai teaches a method, wherein the code portion comprises compiled code for a method corresponding to the method metadata (i.e. *"Computer programs are generally created as source code. The source code is then compiled into object code for execution. Programs generally exist as compiled object code in computer systems. The compiled code is usually designed to operate on only one particular operating system or on only one particular computer processor architecture. In order to use a certain program on several different computer architectures, the original source code must be compiled into object code for each different operating system and each different computer processor architecture."*)(Column 1, lines 18-27).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Adl-Tabatabai to include a method, wherein the code portion comprises compiled code for a method corresponding to the method metadata with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claims 8 and 22, Boger teaches a method in querying the code portion (i.e. *"The primary task of a query optimizer is to choose the most efficient way to execute each database query, or request, passed to the database management system by a user." "An optimizer is often permitted to rewrite a query (or portion of it) into any equivalent form, and since for any given query there* *are typically many equivalent forms, an optimizer has a countably infinite universe of extremely diverse possible solutions (plans) to consider."*)(Column 1, lines 63-66; Column 2, lines 17-22).

Boger does not explicitly teach a method, wherein the code portion comprises searching at N-aligned addresses of the code portion.

Adl-Tabatabai teaches a method, wherein the code portion comprises searching at N-aligned addresses of the code portion (i.e. *"The memory space 400 in FIG. 4a comprises of a run-time stack 410, registers 450, static variables (462, 468, 472) and objects (460, 464, 466, 470) on the heap. The run-time stack 410 may further comprise of a plurality of activation frames (420, 430, 440) for various program functions and methods. These activation frames are used as working space for functions and methods called during execution of the Java program. During program execution, numerous objects or variables may be declared and used within the program. An object, also referred to as a cell or node, is a run-time notion; any object is an instance of a certain class, created at execution time and made of a number of fields. An object may be a dynamically created class instance or an array or an individually allocated piece of data. An object is assumed to be a contiguous array of bytes or words, divided into fields. A field may contain a pointer or a non-pointer value."*)(Column 4, lines 58-67; column 5, lines 1-5).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Adl-Tabatabai to include a method, wherein the code portion comprises searching at N-aligned addresses of the code portion with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claim 9, Boger teaches a method wherein storing the method metadata comprises storing the method metadata (i.e. *"The embodiments discussed hereinafter utilize a metadata manager to facilitate the collection of metadata from a database, and to supply such metadata* *to a plurality of optimizer instances in connection with the optimization of database queries." "As an additional matter, by implementing metadata handling functionality outside of optimizer instances, improvements to metadata collection, refinement and maintenance algorithms may be implemented within a metadata manager without requiring modification of optimizer program code."*)(Column 4, lines 50-54; lines 64-67; column 5, lines 1-2).

Boger does not explicitly teach a method, wherein storing the method metadata at an opposite side of a boundary location at an N-aligned address of the code portion at which a basic block is stored.

Adl-Tabatabai teaches a method, wherein storing the method metadata at an opposite side of a boundary location at an N-aligned address of the code portion at which a basic block is stored (i.e. *"The memory space 400 in FIG. 4a comprises of a run-time stack 410, registers 450, static variables (462, 468, 472) and objects (460, 464, 466, 470) on the heap. The run-time stack 410 may further comprise of a plurality of activation frames (420, 430, 440) for various program functions and methods. These activation frames are used as working space for functions and methods called during execution of the Java program. During program execution, numerous objects or variables may be declared and used within the program. An object, also referred to as a cell or node, is a run-time notion; any object is an instance of a certain class, created at execution time and made of a number of fields. An object may be a dynamically created class instance or an array or an individually allocated piece of data. An object is assumed to be a contiguous array of bytes or words, divided into fields. A field may contain a pointer or a non-pointer value."*)(Column 4, lines 58-67; column 5, lines 1-5).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Adl-Tabatabai to include a method, wherein storing the method metadata at an opposite side of a boundary location at an N-aligned address of the code portion at which a basic block is stored with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

11. Claims 17 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boger et al (U.S. Patent 6,996,556 and known hereinafter as Boger) in view of Buhrke et al (U.S. Patent 5,806,029 and known hereinafter as Buhrke).

As per claim 17, Boger does not explicitly teach a method, wherein searching the code portion comprises bidirectionally searching the code portion for the method bundle.

Buhrke teaches a method, wherein searching the code portion comprises bidirectionally searching the code portion for the method bundle (i.e. *"This is even more problematic for an N-best decoding scheme using a bidirectional search where the backward search should maintain the context dependency used in the forward search in order to achieve search consistency."*)(Column 4, lines 2-6).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Buhrke to include a method, wherein searching the code portion comprises bidirectionally searching the code portion for the method bundle with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claim 28, Boger does not explicitly teach a system wherein the memory further comprises instructions that if executed enable the system to search for the method metadata using one of a forward search, a backward search, or a bidirectional search.

Buhrke teaches a system wherein the memory further comprises instructions that if executed enable the system to search for the method metadata using one of a forward search, a backward search, or a bidirectional search (i.e. *"This is even more problematic for an N-best decoding scheme using a bidirectional search where the backward search should maintain the context dependency used in the forward search in order to achieve search consistency."*)(Column 4, lines 2-6).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Buhrke to include a system wherein the memory further comprises instructions that if executed enable the system to search for the method metadata using one of a forward search, a backward search, or a bidirectional search with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Farhan M. Syed whose telephone number is 571-272-7191. The examiner can normally be reached on 8:30AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Gaffin can be reached on 571-272-4146. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

FMS

JEFFREY GAFFIN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100